

CLAIMS

1 1. A method in a computer system for dispatching requests to perform
2 services to sub-applications that use different logic models the method comprising:
3 providing a context for the sub-applications
4 receiving a request to perform a service; and
5 for a plurality of sub-applications,
6 determining whether the received request should be dispatched to the
7 sub-application; and
8 when it is determined that the request should be dispatched to the sub-
9 application, invoking a service routine of the sub-application passing the request
10 whereby the sub-applications share the provided context.

1 2. The method of claim 1 wherein the sub-applications are ordered and the
2 invoking of the service routines is performed in the order of the sub-applications.

1 3. The method of claim 1 wherein the determining includes determining
2 whether a match criteria for the sub-application matches the received request.

1 4. The method of claim 3 wherein the requests are HTTP requests with a
2 URL and the match criteria is a regular expression relating to the URL.

1 5. The method of claim 1 including suppressing the invoking of additional
2 service routines when an invoked service routine returns an indication to suppress the
3 invoking of additional service routine.

1 6. The method of claim 1 including suppressing the invoking of additional
2 service routines when an invoked service routine responds to the received request.

1 7. The method of claim 1 wherein an invoked service routine performs
2 user authentication and indicates to suppress invoking of additional service routines when a
3 user cannot be authenticated.

1 8. The method of claim 1 wherein an invoked service routine logs the
2 received request.

1 9. The method of claim 1 wherein an invoked service routine logs a
2 response of another invoked service routine.

1 10. The method of claim 1 wherein an invoked service routine transforms
2 the received request from one protocol to another protocol.

1 11. The method of claim 1 including:
2 for each of a plurality of sub-applications,
3 retrieving initialization parameters for the sub-application;
4 retrieving an indication of a class for the sub-application; and
5 instantiating an instance of the class with the retrieved initialization
6 parameters.

1 12. The method of claim 1 wherein the determining includes determining
2 whether a match criteria in a configuration file for the sub-application matches the received
3 request.

1 13. The method of claim 1 wherein the determining is performed by a match
2 routine of the sub-application.

1 14. The method of claim 1 wherein a sub-application uses interaction-based
2 model.

1 15. The method of claim 1 wherein a sub-application uses an action-view
2 model.

1 16. The method of claim 1 wherein a sub-application uses a workflow-based
2 model.

1 17. The method of claim 1 wherein the sub-applications form an overall
2 application and wherein the provided context is an application-level context.

1 18. The method of claim 1 wherein the sub-applications form an overall
2 application that is web-based.

1 19. The method of claim 1 wherein the request is received from a web-
2 server environment.

1 20. A computer system for dispatching HTTP requests to sub-applications,
2 comprising:

3 a configuration file having a class, initialization parameters, and a match
4 criteria associated with the sub-applications;

5 an initialization component that instantiates an object of the class for each sub-
6 application in the configuration file, the instantiated object being initialized with the
7 initialization parameters for the sub-application and being provided with a context object, the
8 context object being shared by the instantiated objects so that the sub-applications share a
9 common context; and

10 a dispatcher that receives HTTP requests and, when the received HTTP request
11 matches a match criteria of a sub-application, invokes a service routine of the instantiated
12 object of the class associated with the sub-application.

1 21. The computer system of claim 20 wherein the match criteria is a regular
2 expression relating to a URL of the HTTP request.

1 22. The computer system of claim 20 wherein the configuration file
2 specifies an ordering of the sub-applications and the dispatcher invokes the service routines
3 in the specified order.

1 23. The computer system of claim 20 wherein the dispatcher does not
2 invoke any additional service routines when an invoked service routine returns an indication
3 to not invoke any additional service routines.

1 24. The computer system of claim 20 wherein the dispatcher does not
2 invoke any additional service routines when an invoked service routine responds to the
3 received request.

1 25. The computer system of claim 20 wherein a sub-application is based on
2 an interaction model.

1 26. The computer system of claim 20 wherein a sub-application is based on
2 an action-view model.

1 27. The computer system of claim 20 wherein each of the sub-applications
2 implement the same interface.

1 28. A computer system for processing request messages, comprising:
2 a plurality of sub-applications forming an application, a sub-application having
3 a match criteria indicating when the sub-application should process a request and having a
4 service routine to invoke when the match criteria indicates that the sub-application should
5 process the request, the sub-applications using disparate logic models;
6 a context for the application that is shared by the sub-applications; and
7 a dispatcher that receives requests, evaluates the match criteria to identify
8 which sub-applications should process the requests, and invokes the service routines of the
9 identified sub-applications wherein an invoked sub-applications use the context.

1 29. The computer system of claim 28 including an initialization component
2 that instantiates an object of a specified class for each sub-application.

1 30. The computer system of claim 29 wherein the initialization component
2 accesses configuration information that specifies the class of each sub-application and any
3 initialization parameters for the sub-applications.

1 31. The computer system of claim 29 including a context object
2 representing the context and wherein the initialization component provides the context object
3 to each sub-application.

1 32. The computer system of claim 28 wherein each service routine is passed
2 a request parameter and returns a response parameter.

1 33. The computer system of claim 28 wherein the sub-applications are
2 ordered and the dispatcher invokes the service routines based on the order of the sub-
3 applications.

1 34. The computer system of claim 33 wherein an invoked service routine
2 indicates that additional service routines should not be invoked to process the received
3 request.

1 35. The computer system of claim 33 wherein the dispatcher does not
2 invoke additional service routines when an invoked service routine responds to a received
3 request.

1 36. The computer system of claim 28 wherein the request is an HTTP
2 request message.

1 37. A computer system for processing request messages, comprising:
2 a plurality of service means for servicing requests, the service means forming
3 an application, each service means having a match criteria indicating when the service means
4 should be invoked, the service means implementing different logic models; and
5 dispatch means for receiving requests, evaluating match criteria to identify
6 which service means should be invoked to process the requests, and invoking the identified
7 service means whereby the service means share a context that is common to the service
8 means of the application.

1 38. The computer system of claim 37 including an initialization means for
2 instantiating an object of a specified class for each service routine.

1 39. The computer system of claim 38 wherein the initialization means
2 accesses configuration information that specifies the class of each service means and any
3 initialization parameters for the service means.

1 40. The computer system of claim 37 wherein each service means is passed
2 a request parameter and returns a response parameter.

1 41. The computer system of claim 37 wherein the service means are ordered
2 and the dispatch means invokes the service means based on their order.

1 42. The computer system of claim 41 wherein an invoked service means
2 indicates that additional service means should not be invoked to process the received request.

1 43. The computer system of claim 41 wherein the dispatch means does not
2 invoke additional service means when an invoked service means responds to a received
3 request.

1 44. A computer-readable medium for controlling a computer system to
2 dispatch requests to perform services to service routines, by a method comprising:
3 receiving a request to perform a service; and
4 for a plurality of service routines,
5 retrieving a match criteria for the service routine;
6 determining whether the received request matches the retrieved match
7 criteria;
8 when it is determined that the request matches the retrieved match
9 criteria, invoking the service routine for processing of the received request
10 whereby the service routines form an application and share a common
11 context.

1 45. The computer-readable medium of claim 44 wherein the service routines
2 are ordered and the invoking is performed in the order of the service routines.

1 46. The computer-readable medium of claim 44 wherein the requests are
2 HTTP requests with a URL and the match criteria is a regular expression relating to the
3 URL.

1 47. The computer-readable medium of claim 44 including suppressing the
2 invoking of additional service routines when an invoked service routine returns an indication
3 to suppress the invoking of additional service routine.

1 48. The computer-readable medium of claim 44 including suppressing the
2 invoking of additional service routines when an invoked service routine responds to the
3 received request.